

①9 RÉPUBLIQUE FRANÇAISE
INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE
PARIS

①1 N° de publication :
(à n'utiliser que pour les
commandes de reproduction)

2 800 889

②1 N° d'enregistrement national : **00 13721**

⑤1 Int Cl⁷ : G 06 F 17/60, G 06 F 9/44

⑫

DEMANDE DE BREVET D'INVENTION

A1

②2 Date de dépôt : 26.10.00.

③0 Priorité : 26.10.99 GB 09925175.

④3 Date de mise à la disposition du public de la
demande : 11.05.01 Bulletin 01/19.

⑤6 Liste des documents cités dans le rapport de
recherche préliminaire : *Ce dernier n'a pas été
établi à la date de publication de la demande.*

⑥0 Références à d'autres documents nationaux
apparentés :

⑦1 Demandeur(s) : MITEL CORPORATION — CA.

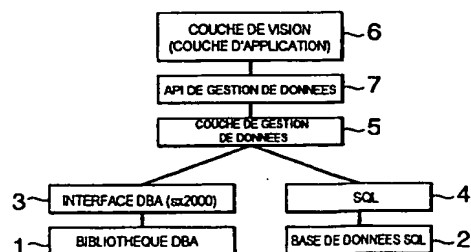
⑦2 Inventeur(s) : MACISSAC BRIAN et SIMSER DAVID.

⑦3 Titulaire(s) :

⑦4 Mandataire(s) : CABINET BEAU DE LOMENIE.

⑤4 **MODELE DE DONNEES COMMUN INCLUANT DES INTERDEPENDANCES DE ZONES.**

⑤7 Un procédé est décrit pour gérer des données à partir d'une certaine variété de sources (1, 2), chaque source disposant de sa propre interface de programmation d'application ou API (3, 4). Une couche de gestion de données (5) qui contient des métadonnées décrit des entités dans chaque source en collationnant les métadonnées et en les présentant sous forme générique. Un contrôleur de vision de niveau supérieur (6) communique avec la couche de gestion de données (5) afin de gérer les données à partir des différentes sources (1, 2).



La présente invention concerne un procédé permettant de gérer des données à partir d'une certaine variété de sources. Il existe des situations dans lesquelles des données existent dans de multiples mémoires ou stockages de données dont chacune ou chacun dispose de sa propre interface de programmation d'application (API), laquelle interface de programmation d'application est un jeu de fonctions disponibles pour que le programmeur les utilise afin de programmer des applications et où il est souhaitable de disposer de la vision générique afférente des données à partir des divers stockages. Un exemple typique de cette situation se rencontre dans le domaine de la téléphonie, domaine dans lequel les stockages de données sous-jacents contiennent une information de téléphonie telle que des numéros de téléphone, des groupes de recherche de ligne et similaire.

Typiquement, chaque source de données qui peut inclure des bases de données commerciales dispose de sa propre API et il n'est pas possible d'assurer un accès générique aux données dans le stockage sans que les couches supérieures disposent d'une connaissance spécifique de l'API de chaque source de données individuelle. Ceci rend les couches supérieures complexes à mettre en œuvre.

Un objet de l'invention consiste à alléger ce problème.

Conformément à la présente invention, on propose un procédé permettant de gérer des données à partir d'une certaine variété de sources, chaque source disposant de sa propre interface de programmation d'application ou API, comprenant les étapes consistant à prévoir une couche de gestion de données qui contient des entités de description de métadonnées dans chaque source, ladite couche de gestion de données collationnant lesdites métadonnées et les présentant sous forme générique, et à fournir un contrôleur de vision de niveau supérieur qui communique avec ladite couche de gestion de données afin de gérer les données à partir de différentes sources desdites sources.

L'invention prévoit par conséquent une couche abrégée de sources de données arbitraires en décrivant un modèle de données générique et une API représentant le moyen pour accéder aux données. Le modèle de données est essentiellement constitué de manière à
5 assurer une description des données qui soit conviviale pour une interface utilisateur.

L'invention prévoit une description de métadonnées générique et une API associée ainsi qu'une représentation générique pour des interdépendances de zones. Les métadonnées sont bien entendu des
10 données concernant les données.

L'invention est flexible en ce sens qu'elle permet l'addition de nouveaux types de zone, elle assure la capacité à décrire des structures de données davantage complexes ainsi que la capacité à décrire des interdépendances de zones davantage complexes qu'au préalable.

15 L'invention peut être utilisée par exemple afin d'assurer un accès à des bases de données produit de Mitel incluant par exemple un symbole, iPBX, une nouvelle commande d'appel etc. Elle peut être mise en œuvre sur un ordinateur à usage général programmé de façon appropriée dans ce but.

20 La présente invention assure par conséquent une vision générique des données à partir de divers stockages de données de même qu'elle permet de constituer une infrastructure pour gérer des stockages de données futurs incluant des bases de données commerciales. L'invention assure un accès générique aux données sans
25 que les couches supérieures aient besoin d'une connaissance spécifique de l'API de chaque stockage de données individuel.

L'invention sera maintenant décrite de manière davantage détaillée à titre d'exemple seulement par report aux dessins annexés parmi lesquels :

30 la figure 1 est un schéma fonctionnel qui illustre la structure du modèle de données conformément à la présente invention ; et

la figure 2 est un schéma qui illustre la représentation générique d'interdépendances de zones.

Sur la figure 1, le modèle générique comprend des sources de données 1 et 2, chaque source de données comportant des interfaces respectives 3, 4. Dans cet exemple, la source de données 1 est une bibliothèque DBA privée et la source de données 2 est une base de données SQL. L'interface 3 constitue une API pour la bibliothèque DBA 1 et un serveur SQL 4 constitue une interface pour la base de données SQL 2.

Les interfaces 3 et 4 communiquent avec une couche de gestion de données, laquelle couche de gestion de données communique à son tour avec une couche de vision 6 par l'intermédiaire d'une API de gestion de données 7. L'API de gestion de données 7 constitue un jeu de fonctions que le contrôleur de vision de niveau supérieur 6 peut utiliser pour gérer les données en provenance des différentes sources 1, 2. Ces fonctions incluent ce qui suit :

- 15 commande de session (connexion)
- obtention d'une liste de visions
- obtention de descriptions de visions
- obtention de descriptions de zones
- transposition d'énumération (dépendant du langage)
- 20 obtention de dépendances de zones
- fonctions de lecture/écriture de données
- commande de transaction

La couche de gestion de données 5 contient les métadonnées pour décrire des entités telles que des produits, des visions, des zones et des interdépendances de zones pour chaque source de données qu'elle gère. Ces métadonnées sont constituées par les systèmes d'accès aux bases de données sous-jacents pour les diverses sources de données. La responsabilité essentielle de la couche de gestion de données consiste à collationner les métadonnées et à les présenter aux couches supérieures sous forme générique. La couche de gestion de données contient des métadonnées pour décrire les entités qui suivent :

produits - La couche de gestion de données ou DML doit pouvoir réaliser une différenciation entre les divers types de produits.

visions (lignes/rangées) - Une vision est une collection lisible/inscriptible de zones à partir d'une base de données. Une vision contient des zones qui peuvent être visualisées par un utilisateur de telle sorte qu'une application puisse afficher toutes les zones à l'intérieur de la vision d'une manière qui prenne sens pour l'utilisateur. Une vision peut contenir des zones qui sont à lecture seulement. C'est-à-dire qu'une zone peut être affichée sans être éditée par un utilisateur.

zones - Une zone est l'entité la plus petite à l'intérieur d'une base de données. Elle doit pouvoir être représentée par un type de données natif (c'est-à-dire qu'elle ne peut pas représenter une structure - par exemple une liste).

groupes de zones - Il s'agit d'un groupage logique de deux zones ou plus à l'intérieur d'une unique vision (par exemple PLID). Le groupage est logique et il n'impose pas un concept d'affichage à l'interface utilisateur graphique ou la GUI.

interdépendances de zones - Il s'agit d'une liste de règles qu'une application peut suivre afin d'assurer un guidage plus important pour un utilisateur.

Les données sont représentées à l'intérieur de la couche de gestion de données selon un format natif. Les couches supérieures disposent de l'option d'extraire le format natif de zones ou de représentations sous forme de chaînes du format natif.

Puisque la couche de gestion de données gère plusieurs sources de données, les descriptions des visions et des zones sont spécifiques au produit. Les couches d'accès aux données de niveau inférieur (par exemple DBA) doivent prévoir une description du type de produit qui est géré par ces couches. A ce niveau, la description de produit peut simplement être une valeur d'énumération de produit. Par exemple :

types de produit
> SX2000 léger
> Symbole
> IPBX

La couche de vision 6 procure une vision qui est une collection lisible/inscriptible de zones à partir d'une base de données. Une vision contient des zones qui peuvent être visionnées par l'utilisateur de telle sorte qu'une application puisse afficher toutes les zones à l'intérieur de la vision d'une manière qui prenne sens pour l'utilisateur. Une vision peut contenir des zones qui sont à lecture seulement. C'est-à-dire qu'une zone peut être affichée sans être éditée par un utilisateur.

Une vision dispose des attributs qui suivent :

10 nom de vision - Une chaîne qui peut être affichée dans une GUI qui identifie la vision.

ID de vision - Une valeur entière qui identifie de façon unique la vision à l'intérieur de la base de données.

Une ID de vision n'a pas besoin d'être unique pour la totalité de la couche de gestion de données ou DML.

15 type de vision

liste de zones - Liste de zones contenues à l'intérieur de la vision opération supportée

liste de groupes de zones

vision parent

20 vision enfant

L'attribut de type de vision est une énumération extensible de types de vision. Une vision doit être de l'un (et de seulement l'un) des types qui suivent :

normal - Pas l'un des autres.

25 singleton - Seulement une ligne/rangée de la vision existe à l'intérieur de la base de données (par exemple des options système)

liste comprimée - Une zone clé à l'intérieur de la vision peut être modifiée au moyen de la base de données sous-jacente lorsque des lignes/rangées sont additionnées et supprimées

30 liste fixe - Aucun enregistrement ne peut être ni additionné, ni supprimé. Un nombre fixe d'enregistrements sont alloués par le système sous-jacent.

Un attribut d'opération supportée énumère l'opération autorisée pour la vision particulière. Une vision peut supporter une ou plusieurs des opérations qui suivent (ou aucune, d'où ainsi la définition d'une vision à lecture seulement) :

- 5 insertion (addition)
- suppression
- mise à jour (modification)

- Dans certains cas, deux visions peuvent présenter une relation parent-enfant. Par exemple, la forme groupe de recherche de ligne dans le SX2000 de Mitel est constituée par une vision d'information et par une vision de données. Cette relation est décrite par des attributs de vision parent et enfant. Si une vision dispose d'une vision enfant, alors l'attribut de vision enfant contient l'ID de vision de la vision enfant. Si une vision dispose d'une vision parent, alors l'attribut de vision parent contient l'ID de vision de la vision parent. Une vision peut disposer de seulement un enfant ou un parent et ne peut pas être à la fois un enfant et un parent.

Une zone est l'unité la plus petite à l'intérieur de la base de données. Elle doit pouvoir être représentée par un type de données natif.

Une zone est décrite au moyen des attributs qui suivent :

- 20 type de zone
- catégorie de zone
- ID de zone
- dimension (longueur)
- valeurs minimum/maximum (Min/Max)
- 25 nom de zone
- transposition d'énumération
- interdépendances de zones

- Un attribut de type de zone est une énumération extensible de différents types. Une zone doit être de l'un (et de seulement l'un) des types de zone qui suivent :

- chaîne arbitraire - Pas de limitation sur les caractères autorisés
- chaîne de téléphonie - Seulement les caractères 0 à 9, * et # sont autorisés

chaîne numérique de téléphonie - les caractères 0 à 9, * et # sont autorisés ou le premier caractère peut être T suivi par les caractères 0 à 9.

5 chaîne de noms - Tous les caractères sont autorisés (a-z, A-Z, 0-9, ! @ # \$ % & () □ < > .) moyennant les exceptions qui suivent : une virgule "," est autorisée et elle est interprétée en tant que délimiteur entre le premier nom et le dernier nom et elle n'est pas incluse en tant que chaîne réelle.

10 La barre verticale "|" n'est pas autorisée du fait qu'elle est utilisée en tant que délimiteur dans z300. La longueur de chaîne ne peut pas excéder le nombre de caractères maximum défini par l'attribut de dimension de zone sans inclure un délimiteur sous forme de virgule.

15 chaîne alphabétique - Seulement a-z, A-Z (pas de valeurs numériques). (Présentement non utilisé sur sx2k mais présenté ici à titre de cohérence).

chaîne numérique - Seulement les caractères 0-9 sont autorisés.

chaîne alphanumérique - Seulement a-z, A-Z, 0-9. Aucun caractère de ponctuation.

entier

20 énumération

jeu symbolique - Equivalent à des jeux de Pascal. Ce type est pour une chaîne basée sur des éléments établis (par exemple des jours de la semaine, lundi, mercredi, mardi à jeudi)

25 jeu numérique - Equivalent à des jeux de Pascal. Ce type est pour des valeurs numériques basées sur des éléments établis (par exemple des groupes COR, "2, 4, 5, 19-24, 35").

Un attribut de catégorie de zone est une liste de zones de bits de catégories de zone. Une zone peut être de l'une ou de plusieurs des catégories de zone qui suivent :

30 clé - La zone est une zone de clé dans la ligne/rangée

lecture/écriture - La zone peut être éditée par un utilisateur

lié - La clé qui est liée à une vision parent ou à une vision enfant

clé de substitution - la zone est une clé alternative pour des opérations d'écriture.

A l'intérieur d'une vision, un groupage de zones peut être décrit. Ces groupages sont des groupes logiques de zones à l'intérieur de la vision. Il s'agit d'une indication pour une application du fait que certaines zones sont associées. Un groupe de zones n'impose pas un aperçu et une sensation par interface utilisateur graphique ou GUI spécifique. Des listes de ces groupages de zones sont contenues à l'intérieur de la description de la vision associée.

10 Un groupe de zones est constitué par l'attribut qui suit :
nom de groupe de zones - Une chaîne qui peut être affichée par une GUI.

type de groupe

15 liste des ID de zones qui sont dans le groupe (ou de pointeurs sur les descriptions de zone)

Un attribut de type de groupe est une énumération extensible de types de groupe. Un groupe doit être de l'un des types qui suivent :

ordonné - L'ordre des ID des zones à l'intérieur du groupe représente un ordonnancement logique des zones (par exemple PLID signifie cabinet, étagère, fente, circuit dans cet ordre). Il s'agit d'un repère pour une application qui peut indiquer un aperçu et une sensation par GUI.

non ordonné - L'ordre des zones dans le groupe est sans importance.

25 Les interdépendances de zones sont des règles qu'une GUI peut souhaiter voir respectées lors de la gestion d'actions utilisateur. Lorsqu'un utilisateur est en train d'opérer sur une zone, l'application peut vouloir avoir connaissance de si les actions de l'utilisateur affectent d'autres zones. Dans ce contexte, une zone peut avoir un impact sur d'autres zones à l'intérieur de la même vision. L'application a besoin de connaître deux choses, c'est-à-dire qu'elle a besoin de connaître quelles autres zones une zone particulière affecte et quelle action il convient de prendre.

Afin de décrire des interdépendances de zones, chaque zone à l'intérieur d'une vision peut disposer d'une liste d'impacts de zone.

Une liste d'impacts de zone contient un ou plusieurs éléments d'impact de zone. Un élément d'impact de zone contient les éléments qui
5 suivent :

ID de la zone soumise à impact - L'ID de la zone qui est soumise à impact par cette zone

condition de test - Une valeur d'énumération de la condition à tester pour déterminer l'action appropriée.

10 contrainte de condition de test - Un pointeur sur une contrainte pour le test

action si "vrai" - L'action à prendre si la condition de test est vraie

action si paramètre "vrai" - Un pointeur sur un paramètre pour l'action vraie

15 action si "faux" - L'action à prendre si la condition de test est fausse

action si paramètre "faux" - Un pointeur sur un paramètre pour l'action fausse.

20 Une énumération de conditions de test est constituée par ce qui suit :

est nulle - La zone contient une valeur nulle. La contrainte de condition de test est nulle.

est au-dessus de la plage - Ce test peut être appliqué pour des valeurs d'énumération ou entières seulement.

25 La condition de test est vraie si la valeur de zone est au-dessus de la valeur pointée par la contrainte de test. Dans le cas d'une valeur d'énumération, la contrainte de test pointe sur une valeur qui est la plus élevée du point de vue de la position à l'intérieur d'une sous-liste de valeurs d'énumération (c'est-à-dire qu'il ne s'agit pas nécessairement de
30 l'énumération de valeur la plus élevée en termes de valeur entière). Dans le cas d'un entier, la contrainte est une valeur maximum.

est au-dessous de la plage - Ce test peut être appliqué pour des valeurs d'énumération ou entières seulement. La condition de test est

vraie si la valeur de zone est au-dessous de la valeur pointée par la contrainte de test. Dans le cas d'une valeur énumérée, la contrainte de test pointe sur une valeur qui est la plus faible du point de vue de la position à l'intérieur d'une sous-liste de valeurs d'énumération (c'est-à-dire qu'il ne s'agit pas nécessairement de l'énumération de valeur la plus faible en termes de valeur entière). Dans le cas d'un entier, la contrainte est une valeur minimum.

est égale - Ce test peut être appliqué pour des valeurs d'énumération ou entières seulement. La condition de test est vraie si la valeur de zone est égale à la valeur pointée par la contrainte de test (pourrait être utilisé pour le type de chaîne également).

est dans la liste - Ce test peut être appliqué pour des valeurs d'énumération seulement. La condition de test est vraie si la valeur de zone est dans la liste pointée par la contrainte de test.

15 Une énumération d'actions est constituée par ce qui suit :

nulle - Poursuivre la vérification

pas d'action

validation - La zone affectée doit être validée

invalidation - La zone affectée doit être invalidée

20 établissement de liste - Peut être appliqué pour des zones d'énumération seulement. Cette action établit l'affectation de zones pour la liste d'énumérations comme pointé par le paramètre d'action associé.

RAZ de valeur - Mettre à zéro la zone affectée

25 établir la valeur - Etablir la valeur de la zone affectée à la valeur pointée par le paramètre d'action associé.

établir la valeur et validation - Etablir la valeur de la zone d'affectation à la valeur pointée par le paramètre d'action associé et valider la zone.

30 établir la valeur et invalidation - Etablir la valeur de la zone d'affectation à la valeur pointée par le paramètre d'action associé et invalider la zone.

On peut par conséquent constater que l'invention propose un modèle de données générique qui permet à l'utilisateur d'accéder à de

multiples sources de données diverses sans la nécessité d'une connaissance spécifique de l'API de chaque source de données.

REVENDEICATIONS

1. Procédé de gestion de données à partir d'une certaine variété de sources (1, 2), chaque source disposant de sa propre interface de programmation d'application ou API (3, 4), caractérisé en ce qu'il comprend les étapes consistant à :

5 prévoir une couche de gestion de données (5) qui contient des entités de description dans chaque source, ladite couche de gestion de données collationnant des métadonnées et les présentant sous forme générique ; et

10 fournir un contrôleur de vision de niveau supérieur (6) qui communique avec ladite couche de gestion de données (5) afin de gérer les données à partir de différentes sources desdites sources.

2. Procédé selon la revendication 1, caractérisé en ce que ledit contrôleur de vision (6) communique avec ladite couche de gestion de données (5) par l'intermédiaire d'une API de gestion de données
15 commune (7).

3. Procédé selon la revendication 1, caractérisé en ce que ladite API de gestion de données (7) prévoit un jeu de fonctions sélectionnées parmi le groupe comprenant : une commande de session, une obtention de liste de visions, une obtention de descriptions de
20 visions, une obtention de descriptions de zones, une transposition d'énumération, une obtention de dépendances de zones, des fonctions de lecture/écriture de données et une commande de transaction.

4. Procédé selon l'une quelconque des revendications 1 à 3, caractérisé en ce que ladite couche de gestion de données (7) contient
25 des métadonnées pour décrire des entités sélectionnées parmi le groupe comprenant : des produits, des visions, des zones, des groupes de zones et des interdépendances de zones.

5. Procédé selon la revendication 4, caractérisé en ce que lesdites métadonnées incluent des interdépendances de zone pour les
30 données sous-jacentes.

6. Procédé selon la revendication 5, caractérisé en ce qu'une zone comporte une liste d'impacts de zone à l'intérieur de chaque vision.

7. Procédé de fourniture d'une couche abrégée pour une certaine variété de sources de données (1, 2), caractérisé en ce qu'il
5 comprend les étapes consistant à fournir une description de métadonnées générique et une API associée pour ladite variété de sources de données et à fournir une représentation générique pour des interdépendances de zones dans lesdites sources de données.

8. Procédé selon la revendication 7, caractérisé en ce que
10 lesdites sources de données (1, 2) contiennent des données de téléphonie.

1/1

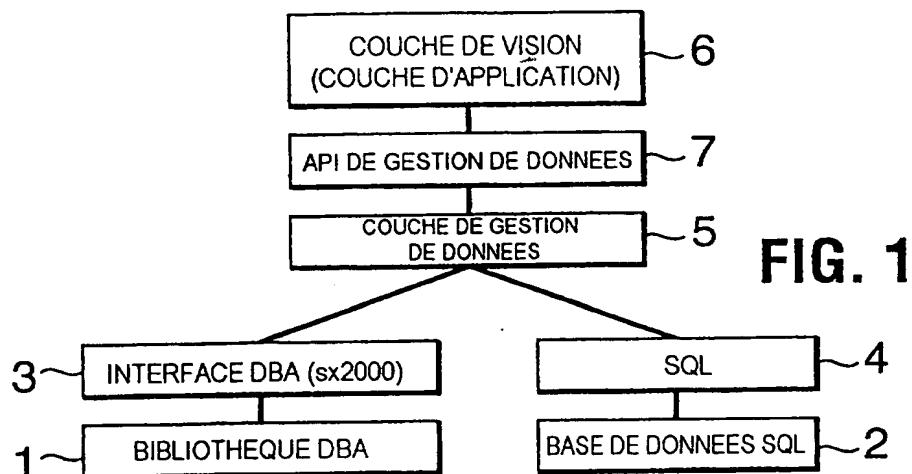


FIG. 1

Descripteur de zone pour zone A

Liste d'impacts de zone *

Liste d'impacts de zone

Zone affectée	Liste d'impacts
B	Liste pour B*
C	Liste pour C*
D	Liste pour D*
<nul>	<nul>

FIG. 2

Liste d'impacts pour B

Condition de test	Contrainte	Action vraie	Param. vrai	Action fausse	Param. faux
est nul	<nul>	Validation	<nul>	<nul>	<nul>
est égal	*valeur	Invalidation	<nul>	Pas d'action	<nul>
<nul>	<nul>	<nul>	<nul>	<nul>	<nul>

Liste d'impacts pour C

Condition de test	Contrainte	Action vraie	Param. vrai	Action fausse	Param. faux
Plage au-dessus	*dernière	Etab. liste	*liste 1	Etab. liste	*liste 2
<nul>	<nul>	<nul>	<nul>	<nul>	<nul>

Liste d'impacts pour D

Condition de test	Contrainte	Action vraie	Param. vrai	Action fausse	Param. faux
est dans liste	*liste	Etab. valeur	*valeur	Pas d'action	<nul>
<nul>	<nul>	<nul>	<nul>	<nul>	<nul>

This Page Blank (uspto)